AD NUMBER
ADB166990
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to DoD and DoD contractors only; Specific Authority; 08 SEP 1992. Other requests shall be referred to Defense Information Systems Agency, 701 South Courthouse Road, Arlington, VA 22030.**
AUTHORITY
OASD dtd, Jul 1994

The Use of IDEF in Functional Economic Analysis

1991

Edgar H. Sibley* +

and

Minder Chen+

Department of Information and Software Systems Engineering*
Department of Decision Sciences and MIS+
George Mason University
Fairfax, VA 22030

Abstract

The use of IDEF is mandated by the DDI (DoD's Director of Defense Information) for definition of all Business Case and Functional Economic Analyses (BCA/FEA). This brief paper reviews the IDEF methodology and considers aspects of its use, both its advantages and disadvantages.

1. Introduction

The use of the IDEF (a system definition methodology) is required by the DDI in documenting parts of required FEA (Functional Economic Analysis). This methodology derives, in its present form, from the Integrated Computer-Aided Manufacturing (ICAM) program, where a specification standard methodology was needed for manufacturing data specification and interchange; a family of tools is available to aid the IDEF process. IDEF stands for: ICAM definition.

A very brief review of the FEA process is given first. Then, because there have been questions about the use of IDEF *versus* certain other methodologies and automated tools, we attempt to answer the following questions:

- What is the IDEF methodology?
- What is the relative value of such a methodology and what are the tools currently available to support them.
- What is the difference between IDEF and the many Information Systems methodologies and their supporting CASE (Computer Aided System Engineering) tools?

2. The FEA and the Systems Design Process

The terms Business Case Analysis (BCA) and FEA are often used interchangeably. Here we attempt to distinguish them with the following definitions.

A BCA is the process of modeling and analyzing the business activities of an organization and its supporting infrastructure.

The use of IDEF in Functional Economic Analysis (FEA)

VOLUME 2a

Task Order 9

Contract # DCA 100-91-C-0033

Prepared For:

The Center for Information Management
Defense Information Systems Agency (DISA)
701 South Courthouse Road
Arlington, VA. 22204

Prepared By:

C3I Center for Excellence
George Mason University
Fairfax, VA. 22030

Obviously, not the least problem in making a BCA is in determining the scope or contour of the business being analyzed: Does it encompass all business or only certain parts? Once a set of specific units or offices have been identified, this is further refined into the questions:

- What interactions are there between this portion of the business and others? and
- How can the parts be isolated so that the boundaries are not confusing to the analysts and the allocation of the savings is not disputed by the various business areas?

FEA is more specific to DoD operations.

An FEA deals with the business analysis of DoD defined functional areas (those under the control of a Functional Manager), using them as a departure point for an initial cut at scoping the area or unit under consideration.

The overall FEA process involves several steps:

1. **Create an initial Business Process Model.** An initial business process model is created by interviews or having users and functional area managers participate in a facilitated business process modeling workshop. Then baseline business process model is created with the help of functional area users, managers, systems analysts, cost analysts, and other experts. A pictorial process model is developed to help visualize the results and understand the analysis.
2. **Determine Activity-Based Performance and Cost Data.** Cost and performance data is determined for the initial process model and used a baseline. A high-level data model for the functional area may also be developed at this stage.
3. **Develop Alternative Business Process Models.** The project team evaluates the baseline business process model to identify opportunities for streamlining and redesigning processes. A set of criteria, such as generalized coupling and cohesion, is developed in evaluating alternatives.
4. **Evaluate Alternative Business Process Models.** Alternative business process models are evaluated based on cost and performance data, such as the investments compared with the baseline. Performance data, such as volume and frequency of inputs and outputs as well as response time of processes is used to build a simulation model to analyze the dynamic behavior of the alternatives. Costs are also derived.
5. **Present and Select One of the Alternatives.** A final set of alternatives is presented to top management, a recommended alternative is identified. Top management will select one alternative.

The term "system" as used in the field of modeling generally implies:

a relatively large complex of interconnected parts with an organized array of individuals forming and working as a unit.

The criteria for choosing a system modeling methodology is primarily that it be appropriate for modeling the type of system under consideration. It must also be:

- Easy to learn and use by its intended users.
- Capture information of the target system in a structured way so that the information can be further analyzed, preferably by computer programs.
- Support decomposition of the system into a hierarchical organization (e.g., a decomposition) of components with different level of abstraction.

3. The IDEF Family of System Design Methodologies

The first of the IDEF methodologies was called $IDEF_0$. This is a functional modeling method, somewhat similar to a conventional procedural and control flow chart with aspects of a general data flow diagram. A second subset ($IDEF_1$) is based on Chen's Entity-Relationship-Attribute model for conceptual data base design. $IDEF_1$ Extended ($IDEF_{1x}$) was developed under the Integrated Information Support System (IISS) project, under the ICAM Program, as a data modeling technique to describe common data model subsystem. The primary contractor is General Electric Company. The final report was delivered in November 1985. This has since been extended, mainly by D. Appleton Company (DACOM). Other members of the IDEF family include a simulation language ($IDEF_2$) developed mainly by Pritsker & Associates. It can be used to simulate a design, thereby representing

time varying behavior of resources in a system. There is an on-going effort to extend the IDEF suite of methods [Mayer and Painter, 1991], as summarized in Table 1. Other members of the IDEF family ($IDEF_7$ to $IDEF_{14}$) are in review by the IDEF Users Group. In this report, the discussion is focused on $IDEF_0$ and $IDEF_{1x}$ and their applications in BCA/FEA.

IDEF Method	Description	Document or Status
$IDEF_0$	Functional modeling method derived from SADT.	ICAM program report, 1981.
$IDEF_1$	Information modeling method derived from Peter Chen's, Entity-Relationship model,	ICAM program report, 1981.
$IDEF_{1x}$	Data modeling method that support logical design of relational databases.	ICAM Project report, 1985.
$IDEF_2$	Dynamic modeling method for describing time varying behavior of functions and information.	ICAM Program report, 1981.
$IDEF_3$	Process flow and object state description method that uses a scenario driven process to capture domain expert knowledge about the behavior of a system.	Emerging
$IDEF_4$	Object-oriented design method to assist design of a system implemented in object-oriented technology.	Emerging
$IDEF_5$	Concept/Ontology description method used for knowledge acquisition in building knowledge-based systems	Emerging
$IDEF_6$	A method that facilitates acquisition, representation, and manipulation of design rationale of a system.	Still in a formative stage

Table 1. A Summary of the IDEF Suite of Methods

A model is expected to answer questions about the requirements and initial design decisions of a system. It should have a clearly defined boundary, a viewpoint, and a purpose.

Data is the foundation of modern information systems enabled by data base technologies and it should be managed as a corporate-wide resource. The basic assumption of data modeling is that data in an organization exist and can be described independently of how these data are used. The types of data used in an organization do not change very much and they have certain inherent properties which lead to correct structuring of a data model which serves as a foundation for the development of an enterprise-wide/function-specific database system. A well-designed data model should be a stable one. A data modeling technique used in the BCA/FEA process should be

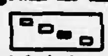
a conceptual data modeling technique which uses a minimum set of graphical notations to support the design of end-user oriented data models.

An *entity* is a class of objects or events, real or abstract, about which we store data. It represents a set of *entity instances* which can be described by the same set of *attributes*. The *attribute value* of the same attribute for each *entity instance* may be different. The name of each *entity* is a noun or an adjective and a noun in the singular form. Examples of entities are product, employee, project, department.

3.1 The $IDEF_0$ Diagram

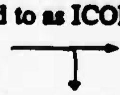
Activities and things are two object types that can be modeled in $IDEF_0$. *Activities* are functions and processes performed by the system. *Things* include both data (e.g., a blueprint, customer orders) and non-data objects (e.g., parts, machine tools, an automated system). The interactions between things and activities are called *connections*. $IDEF_0$ diagrams have boxes and arrows. *Boxes* represent activities and have dominance. *Arrows* represent things that interact with activities they are interconnections between boxes. Arrows connect boxes together to indicate constraints or dominances of one activity on anther.

A $IDEF_0$ diagram also contains the following information:

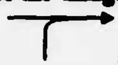
- Title
- Author(s) of the diagram
- Project Name
- Date of creation
- Date of last revision
- Diagram status: Working, Draft, Recommend, or Publication
- A list of readers and dates when read.
- Context: Position of the diagram in the parent diagram, e.g., *None* for A0 diagram, *Top* for first-level decomposition diagram, or  to show the relative position of the parent process in the decomposition diagram to which it belongs.
- Node Number: A unique identifier of a box. It consists of project abbreviation, "/", Node index number of the parent node, and box number of the node in the diagram. For examples: FEA/A1, FEA/A13, FEA/A32.
- Notes: Substantive comments about the diagram
- C-Number: An $IDEF_0$ diagram' unique identifier (e.g., MC 002). It consists of the author's initials (or unique identifier) and a sequence number. The C-number of a previous version of the diagram is enclosed in parentheses to provide link to prior work.

A system can be modeled as an set of interrelated $IDEF_0$ diagrams, texts, and glossary. Diagrams are the major components of the model. The highest-level diagram, called A0, represents the whole system. An A0 diagram has only one box and is annotated by PURPOSE and VIEWPOINT. The PURPOSE is to answer questions about the system; e.g., the reason why the model was developed. A system can be described from several VIEWPOINTs (i.e., that of a person or an organizational unit). An $IDEF_0$ should, however, be developed from only one particular viewpoint.

The role of the arrows is important in $IDEF_0$ diagrams:

- Arrows represent how boxes influence or constrain each other. A box can send its output to another function for further transformation or provide a control that govern what another function must perform.
- The side of the box at which an arrow enters or leaves determine the role of an arrow (i.e., a thing) related to the box. These roles include input (left), control (top), output (right), and mechanism (bottom) They are referred to as ICOM in $IDEF_0$ diagrams.
- Arrow branches  may be one of two types. Branches that are not labeled are assumed to contain all the things carried by the arrow before the branch. If they are labeled, they could contain

some or all of the things carried by the arrow before the branch.

- Arrow joins  are similarly of two types. Branches that are not labeled are assumed to contain all the things carried by the arrow after the join. Branches that are labeled could contain some or all of the things carried by the arrow after the join.

A generic IDEF$_0$ diagram in Figure 1 can be described as:

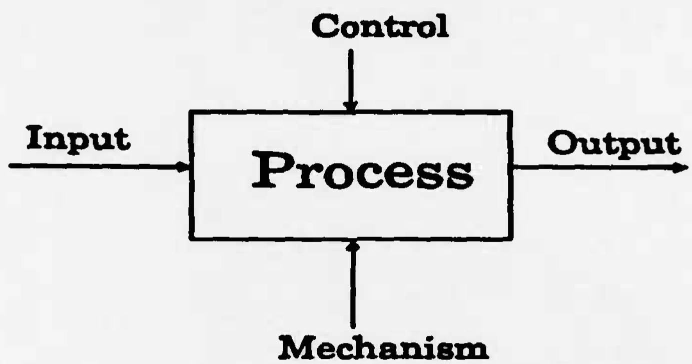
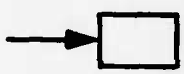
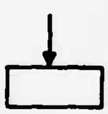
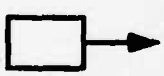


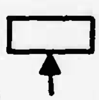
Figure 1. The Basic Constructs of IDEF$_0$ Diagrams

Inputs are transformed by the *function* into *outputs* according to *controls*, using *mechanisms*.

- Inputs  describe resources or data that are needed to perform the function and are transformed by the function into outputs.

- Controls  describe the conditions, rules, procedures, or circumstances that govern the execution of the function. An arrow is a control unless it obviously serves as input. Each function should have at least one control arrow. Most controls are in the form of data.

- Outputs  are the data or non-data objects that are transformed by the process and leave its boundary. An output of one process can be used as inputs, controls, or mechanism of another.

- Mechanisms  define the actors (i.e., supporting mechanisms) that carry out the function. A mechanism may be a person, an organizational unit, a physical device, a computer program, etc.

IDEF$_0$ requires 3 to 6 boxes in any one diagram except the A0 diagram. IDEF$_0$ tends to be used as a physical model because it can describe the implementation *mechanisms* of systems functions. However, IDEF$_0$ can be used to provide a logical (i.e., essential) model of the system. IDEF$_0$ can be used to show material flows. The definition of control in IDEF$_0$ is much broader than the control flow and control process used in real-time Data Flow Diagram.

3.2 *The IDEF$_1$ Diagram*

An entity can be described by its name, description, a set of attributes. Two types of entities may be represented in IDEF$_{1x}$ by using differently shaped boxes: *identifier-independent* and *identifier-dependent* entities. An identifier-independent entity shown in Figure 2(a) is also called an independent entity. Each instance of the independent entity can be uniquely identified without the existence of any relationships with other entities. An identifier-dependent entity is shown in Figure 2(b); it is also termed a dependent entity. The existence and unique identification of each instance of a dependent entity depend on the existence of its relationship to another entity.

Figure 2 Two Types of Entities

A relationship is an association among entity types. In IDEF$_{1x}$ a relationship typically involves two entities (i.e., a binary relationship) or one entity (unary or recursive relationship). A relationship involved more than two entity types (i.e., N-ary relationship) can be represented using an associative entity. Associative entities are not explicitly defined in IDEF$_{1x}$.

An entity-level IDEF$_{1x}$ diagram displays entity names and/or relationship names. Many to many relationships are allowed. An example of entity-level IDEF$_{1x}$ is shown in Figure 3.

Figure 3. Entity-level IDEF$_{1x}$ Diagram

A key-level IDEF$_{1x}$ diagram displays the entity names as well as the primary keys of the entities in the diagram as depicted in Figure 3. Non-specific relationships should be resolved into two specific connection relationships.

Figure 4. Key-level IDEF$_{1x}$ Diagram

An attribute-level IDEF$_{1x}$ diagram displays the entity names, the primary keys and all the non-key attributes of the entities in the diagram as depicted in Figure 5.

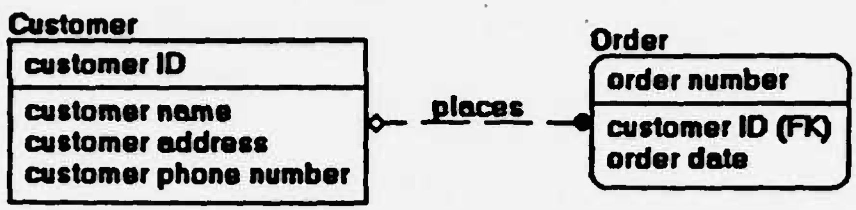


Figure 5. Attribute-level $IDEF_{1x}$ Diagram

A specific connection relationship is also called a connection relationship., which is depicted as a line drawn between the parent entity and the child entity with a dot at the child end of the line. Cardinality of the connection relationship is frequently termed a business rule, because it implies certain business policies and rules. For example, "a manager can only manage one and only one organization unit." The default child cardinality is 0 to 0, 1, or many. A letter/number symbol is used to represent the cardinality of the child entity associated with the relationship.

Specific connection relationships can be classified as: identifying and non-identifying relationships. In an identifying relationship the parent entity is used as part of the primary key of the child entity. The child entity is always an identifier-dependent entity. The parent entity will be an identifier-independent entity unless the parent entity is a child entity of some other identifying relationship. In a non-identifying relationship, the primary key of parent entity is an non-key attribute of its child entity.

A non-specific connection relationship represents a many-to-many relationship between two entities. It is represented as a solid line between two entities with a black dot connects to each of the entity as depicted in Figure 6.

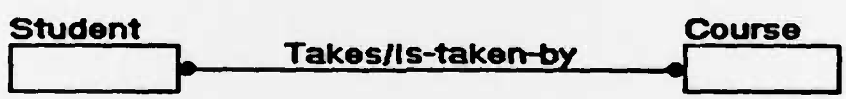


Figure 6. Non-specific Relationship

Categorization Relationship support the generalization/specialization concept in data modeling. A categorization entity is a collection of entities of the same type to which a narrower definition and *additional* attributes and relationships apply. A *categorization entity* inherits (as in an object oriented approach) all the attributes and relationships of its parent entity type (called a *generic entity*). An attribute whose values partition the generic entities into categorization entities is called a *category discriminator*. If each instance of the generic entity can be classified into one of the categorization entities, the categorization relationship is called complete categorization relationship. If some instances of the generic entity cannot be classified into existing categorization entities, the categorization relationship is called incomplete categorization relationship.

Attribute describe entities, but not relationships. Attributes for entities are underlined in following examples:

- Product has <u>quantity-on-hand, weight, volume, color,</u> and <u>name</u>.
- Employee has <u>SSN</u>, <u>salary</u>, and <u>birthday</u>.

An *alternative key* is a unique identifier. It can be a simple attribute or a composite attribute used to identify an instance of an entity. For example, EMPLOYEE may have two candidate keys: Employee ID and SSN

A *Secondary (Indexed) Key* is a key that has been used by the underlying DBMS to create indexes for fast retrieval. A *Concatenated (Combined) Key* is a key that consists of a more than one attributes. A *Foreign Key* is an attribute (a non-key attribute or an attribute that is part of the primary key) in a relation, that has been used as a primary key in another relation. It is also called a *linking field*.

An Associative Entity Type is an Entity Type whose existence is meaningful only if it participates in several (>=2) Relationship Types at the same time. Associative Entity Types are often introduced to represent additional information in *many-to-many* Relationships or to decompose a *many-to-many* Relationship into two *one-to-many* Relationships. Associated Entity Types are also used to represent *n-ary* (n>=3) Relationships in a binary data model.

4. Availability of IDEF and BCA/FEA Support Tools

There are several tools that could be used to support the application of IDEF and BCA/FEA. Many techniques and methods used in an FEA study are complex and a large amount of structured information is generated in the definition process. Selecting appropriate tools and integrating them to support the techniques and methods is critical to the successful implementation of an FEA.

4.1 Computer-Aided Diagramming Tools for $IDEF_0$ and $IDEF_{1x}$

Currently there are several COTS that support $IDEF_0$ and $IDEF_{1x}$. Some basic information of these IDEF tools are listed in Table 2.

Product Name	Methodology supported	Vendor Name	Hardware
IDEFine-0, IDEFine-1x and IDEFcost	$IDEF_0$, $IDEF_{1x}$	Wizdom Systems, Inc	Sun, PC, Apollo, VAX
Design/IDEF	$IDEF_0$	Meta Software Corp.	Mac, PC, Unix workstation
IDEF/Leverage	$IDEF_0$ $IDEF_{1x}$	D. Appleton Company	PC, VAX, microVAX
AI0 & AI1x	$IDEF_0$ $IDEF_{1x}$	Knowledge Based Systems, Inc	PC, LAN
Authormate	$IDEF_0$	Eclectic Solutions	PC, VAX
ERWin	$IDEF_{1x}$	Logic Works	MS Windows

Table 2. IDEF Tools for IDEF0 and 1

There are several approaches that DISA can take in providing computer-aided diagramming tools to support the use of IDEF techniques. All of these may be viable alternatives.

1. Select the best tool from the marketplace and make it a standard tool.
2. Recommend a list of tools that support existing IDEF methods specified in the ICAM program.

3. Recommend use of a CASE shell to generate diagramming tools that support IDEF to meet the specific requirements in applying IDEF for FEA.

One advantage of the latter alternative is that CASE shells provide a flexible approach to supporting existing and future methods required in the BCA/FEA process. CASE shells are software tools that allow users to define a system modeling methodology and then generate a run-time environment to support the specification of a target system using the methodology. There are several commercially available CASE shells in the marketplace. It will be a worthwhile effort for CIM to evaluate these CASE shells. The benefits of the CASE shell is that, it allows the customization and evolution of the methodology. Existing IDEF tools might need appropriate extensions to support the modeling and analysis. Using a CASE shell to generate tools to support IDEF will allow continuous improvement of IDEF and smooth integration of IDEF with tools needed in the BCA/FEA and in the downstream systems development process. Examples of such CASE shells are: ProtoSoft's Paradigm Plus CASE Development Kit, Intersolv's XL/Customizer, Systematica's Virtual Software Factory, CADWare's Sylvia Foundry, etc. However, the Graphics User Interface in many of these tools is still primitive.

The work of certain standards committees, such as the IDEF Users Group and the Electronic Industries Associate's CDIF (CASE Data Interchange Format) Committee may soon make it possible to pass designs from one vendor's tool to another.

4.2 Tools That Support the BCA/FEA

There are several other types of tools that could support the BCA/FEA process. Using IDEF for business process modeling under the context of BCA/FEA is a collaborative effort among functional area users, managers, cost analysts, information engineers, and business engineers. It is a change process because the re-engineering business process may dramatically change how work will be done in the future. Team-oriented techniques, such as Joint Application Design may be used to encourage proper participation of all parties involved in the process to ensure all the requirements and concerns are surfaced and addressed. One of the critical success factors in using JAD is the skill of the facilitator (i.e., session leader). However, skillful facilitators are hard to find. Emerging collaboration technologies can be used to assist facilitate business process modeling workshop. Collaboration technology can be used to provide anonymity, equal participation, and complete documentation of workshop outcomes. It is not going to replace a good facilitator, but it can be used to improve the effectiveness of business process modeling.

Conducting BCA/FEA is a very political process. Group facilitation techniques and collaboration technologies can be used to assist to support business process modeling meetings. Group support systems, such as GroupSystems and VisionQuest can be used as a front-end requirements elicitation tool that capture initial specifications of a business process [Chen and Nunamaker, 1991]. These specifications can be transferred to IDEF tools for the construction of formal IDEF models. We have developed a Collaboration Technology Laboratory at George Mason University, that can be used to demonstrate this approach.

The IDA Template can also be used to present the final cost analysis of various alternative business process models with the baseline model. Cost data collected for various alternatives and for the baseline can be presented in different formats (e.g., graphics and tables) based on a specific cost breakdown structure. The two major cost items are: Operations Costs and Management & Support Cost. Each of them is broken down into the major life cycle phases and expense types. Cost data of the baseline and alternatives are entered into a cost Data Sheet based on risk and the detailed cost breakdown structure over a six-year period. For alternatives, each estimated cost item has the high and low values. It is a way to express the risk or uncertainty involved in the estimation. An analysis is performed on each alternative. The results are presented in graphics and tables.

5. Comparison of IDEF Methodology with Dataflow, ERA, and Other Techniques

A system can be modeled from multiple viewpoints. However, $IDEF_0$ only allows a user to model a system from

a single viewpoint. A system model consists of a set of interrelated diagrams, texts, and tables. Dataflow diagrams (DFD) can be used to describe both the logical and physical aspect of an existing (AS-IS) system or a new (TO-BE) system, just as can $IDEF_0$. When a DFD is used to specify the physical aspect of a system, the implementation mechanisms can be amended to the processes, data flows, and data stores in the diagram. Mechanism can also be captured in the corresponding entry of an object in the repository. Configuration management accessories in many CASE tool environments today should substantially aid in relieving the problems noted above.

Real-time Data Flow Diagramming (DFD) Technique can be used to achieve the same representation power as $IDEF_0$. The real-time versions of DFDs can be used to represent data flow to represent inputs and outputs of a process. Control flow has been amended in the real-time DFD to represent signals that enable and disable data processes. Control processes are used to coordinate the execution of data processes. Control processes can then be augmented by State-Transition Diagram (STD) or Petri Net processors.

The weakness of $IDEF_0$ include:

1. The external entities are not explicitly represented. and thus the interaction of the system with external entities is only represented by the inputs, outputs, controls, and mechanism of the system.
2. Boxes in $IDEF_0$ are forced to be placed as a *staircase* pattern to show the dominance among them. However, there are situations where several functions may have the same dominance. The author of a $IDEF_0$ diagram still has to lay out these functions as if they were different in their dominance.
3. The glossary of $IDEF_0$ is a simple description of the objects used in the diagram. It cannot represent additional attributes of the objects or relationships of the objects with other objects. A more powerful repository is needed to integrate $IDEF_0$ diagram with $IDEF_{1x}$ diagram, and with other modeling tools used in the process.
4. INPUT, OUTPUT, MECHANISM, and CONTROL are terms used to describe how things (i.e., data and materials) are related to a function or activity under study. An output of a function can be used as an input or control of another function. Controls are usually in the form of information. It is not very easy for $IDEF_0$ users to determine the role of a thing that interacts with the function. It may have to be interpreted by the user, who could easily make mistakes.

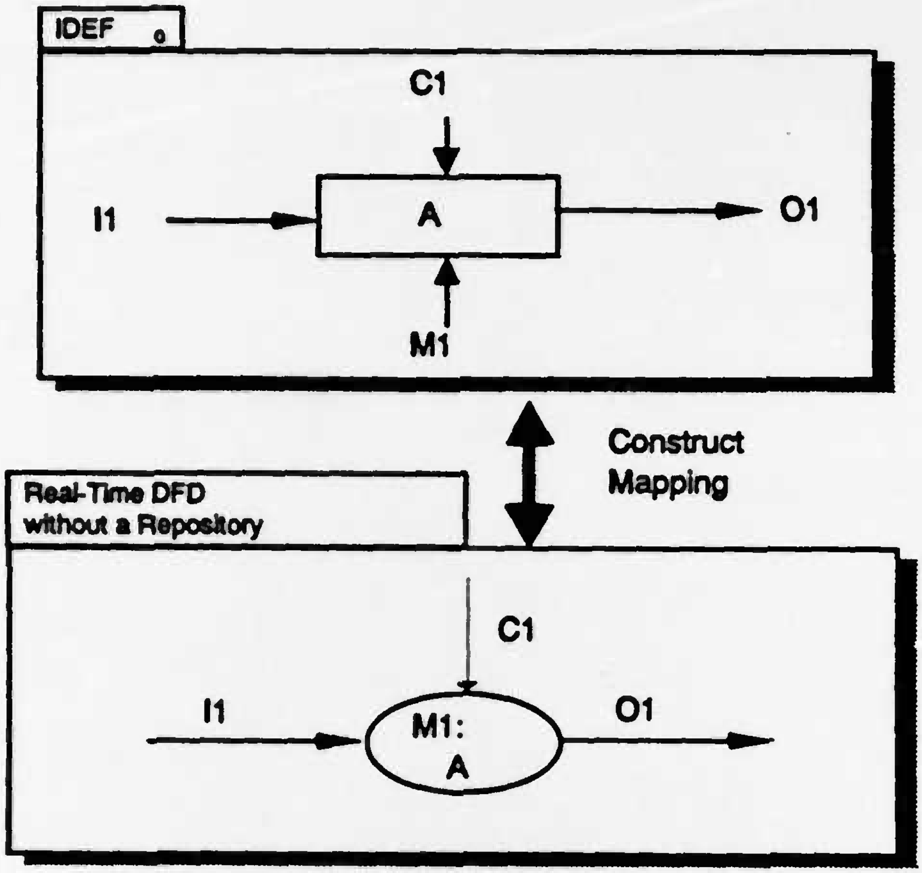


Figure 7. Mapping Between $IDEF_0$ and Real-time DFD

Figures 7 and 8 illustrate the mappings between $IDEF_0$ diagrams and real-time CASE tools

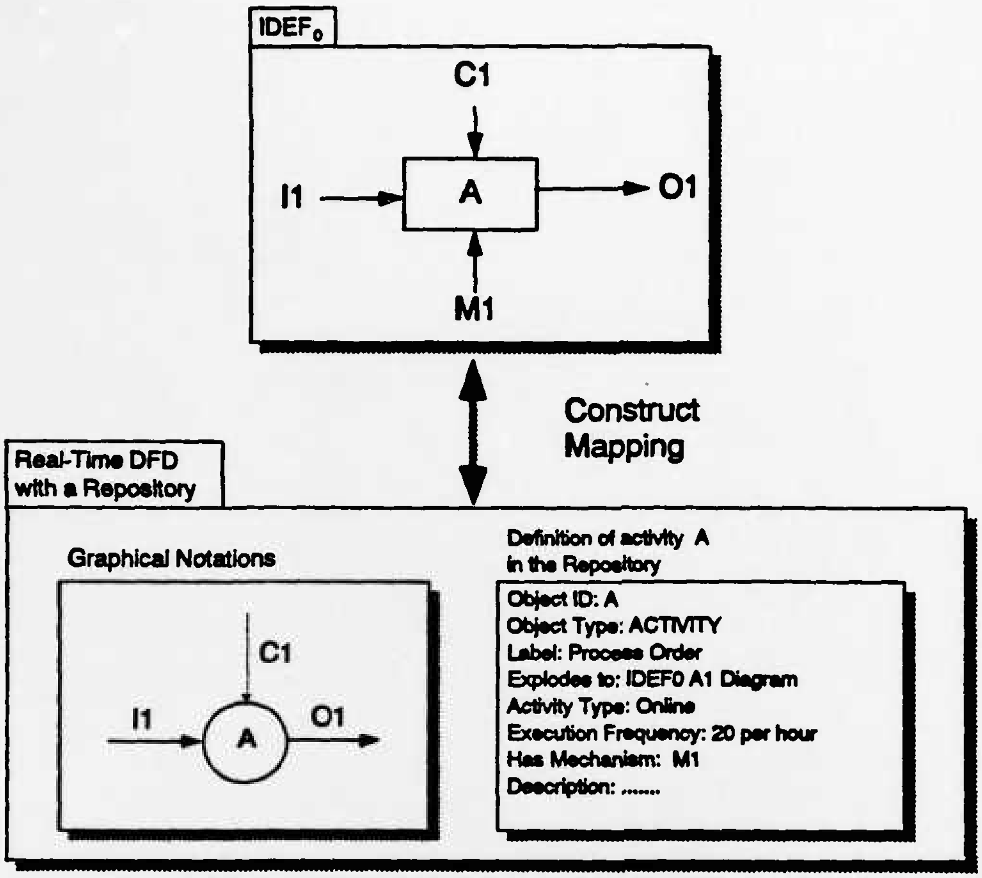


Figure 8. Construct Mapping Between $IDEF_0$ and a Real-time DFD with a Repository

The major weakness of $IDEF_{1x}$ appears to be that it would be inappropriate as a conceptual data modeling tool. and also that the graphical notations seem unnecessarily complex. For examples, the identifier-dependent and identifier-independent entities have different shapes. However, whether an entity is identifier dependent or independent is mainly a logical or physical database design decision. The distinction between identifying and non-identifying relationships is of the same nature. At the conceptual data modelling level, there is no need to make such distinctions. The cardinalities of specific connection relationship and non-specific relationship can be represented by a line connecting two entities augmented by a pair of numbers (representing minimum and maximum) or symbols attached to the connecting entities. The notations of non-specific relationship and specific relationship (with four possible cardinalities) can be reduced to just one concise notation. This would make learning and applying $IDEF_{1x}$ much easier.

The entity relationship model is an end-user oriented data modeling technique that is widely supported in existing COTS CASE tools. There are several variations of ER diagramming notations. ER model is a conceptual data modeling technique. The process of converting an ER diagram to a logical or physical database schema can be computer aided. For BCA/FEA, in which the primary users are from the business functional areas, the data modeling technique should be kept at as high a level as possible. If data modeling is required as part of BCA/FEA, we minimum use of $IDEF_{1x}$ (the use $IDEF_{1x}$ at the entity-with key attribute level only).

We believe, however, that data modeling should be a part of BCA/FEA, because understanding the data structures of a function could help the design and redesign of business processes. Many information systems that support existing business functions do not share data; thus many redundant business processes have been created to reenter and revalidate data already in the system. In the BCA/FEA stage, a data model may help business managers

redesign the business processes according to a *shared* database concept. Many duplications or inefficiencies of existing business processes could then be eliminated.

$IDEF_{1x}$ uses a set of terminologies that are uncommon to the database design community. Users may be confused if $IDEF_{1x}$ is used at the attribute-level, unless they are properly trained in normalization theory. Currently, there is no *repository* concept in IDEF techniques. Objects (e.g., activities, inputs, and outputs) are defined in a glossary that capture only limited attributes of the objects. Structural relationships among objects are not captured. The integration of IDEF suite of methods could be achieved by using a repository system that can support the data integration among tools that support IDEF methods and related tools for BCA/FEA. Currently, IDEF does not provide a predefined set of attributes and relationships for further description of the characteristics of IDEF objects and their relationships to other objects. Due to this deficiency, if IDEF has to be used in BCA/FEA, a meta-model of the IDEF has to be defined to serve as a foundation for implementing repository-based IDEF tools [Chen and Sibley, 1991]. Important information required for BCA/FEA should be defined in the meta-model. The following are some suggestions:

1. Cost and resource utilization data could be associated with MECHANISM. These include volume, frequency, unit cost, etc.
2. Performance data could associated with INPUTS, OUTPUTS, and PROCESSES. Examples of performance data include response time, throughput, etc.
3. Things can be categorized into data and non-data objects that ink data objects used as inputs, controls, or outputs of processes in $IDEF_0$ with $IDEF_{1x}$ diagrams or lwith other data structures descriptions.

6. Using IDEF for Business Process Modeling

In discussing the ISDOS project in retrospect, one author has said that there was major industrial apathy due to a lack of the use of its PSL/PSA generated documents for implementing the target system [Sibley, 1986] -- thus the users saw it only as a complicated documentation device. If the only purpose of using IDEF in BCA/FEA is to document the business process model, the potential benefits of using it will be limited. We believe that IDEF or its alternative techniques should be used for continuous business improvement and for downstream systems development activities. If a technique or tool is used only for documentation, no one is likely to have a vested interest in keeping the model up-to-date. The effort spent in building the business process models may be wasted.

The business expertise embedded in business models is a valuable asset that should be exploited by functional area users and managers with the assistant of business analyst. Once the model has been developed, it could be used to by functional area users and managers to:

1. support continuous business improvement based on the performance criteria established in the model,
2. assist the business reengineering process in making structural changes,
3. navigate and explore the model in order to understand the goals and objectives of the organization,
4. guide the development of IS models to ensure that the IS are aligned with business objectives.
5. perpetuate a common mental model of the business.
6. form the basis for delivering information in the business context.

Business process models should be integrated with information systems models because many critical business functions are supported by IS. An integrated model should be used not only for the design/redesign of business and its information systems, but also the users to deliver the information [Chen, Nunamaker, and Weber, 1989; Chen, Liou, and Weber, 1992].

7. Conclusions

Training is one of the important factor in successfully implementing a methodology. It is particularly critical in implementing IDEF as a front end modeling technique to support business engineering/reengineering and to facilitate BCA/FEA because these processes are driven by end users and managers. They are usually not familiar with the systematic approach in analyzing and designing business systems and information systems. However, their involvement is politically smart. It is the only way to capture the information and knowledge about the

businesses and their operations. By tapping into front-line users' and managers' expertise and creativity about how they may improve their work processes, we will have a better chance to introduce dramatic improvements in the processes. Training on IDEF in the FEA should be given to both business analysts and user/managers of various agencies. *Real-time training* should be given so that people with the training can apply the techniques, methods, and tools in their jobs.

The BCA/FEA is a change process. It changes the way we think about systems investment and development processes and the users' and managers' roles in these process. It is a change of mind set instead of a change of notations. Therefore, it is more important to educate the agencies about the underlying principles of CIM first, then familiarize them with BCA/FEA processes. IDEF and other related techniques, methods, and tools are important mechanisms to carry out the vision of CIM, but they should be introduced after the CIM principles and process have been fully understood.

8. References

NOTE:
These are used in a more complete discussion to be provided with the final deliverable.

1. Brewin, B., "CIM: Ccorporate Information Management," white paper, Federal Computer Week, September 1991, pp. 1-15.
2. Chen, M., Liou, Y. I., and Weber, E. S., "Developing Intelligent Organizations: A Context-Based Approach to Individual and Organizational Effectiveness," to appear in Organizational Computing.
3. Chen, M. and Nunamaker, J. F., Jr., "The Architecture and Design of a Collaborative Environment for Systems Definition," Data Base, Vol. 22, No. 1/2, Winter/Spring 1991, pp. 22-29.
4. Chen, M., Nunamaker, J. F., and Weber, E. S., "The Use of Integrated Organization and Information Systems Models in Building and Delivering Business Application Systems," IEEE Transactions on Knowledge and Data Engineering, Vol. 1, No. 3, 1989, pp. 406-409.
5. Chen, M. and Sibley, E. H., "Using a CASE Based Repository for Systems Integration," The 24th Annual Hawaii International Conference on System Sciences, January 8-11, 1991.
6. Cost-Based Activity Modeling Project Results, Proposal #91-20, presented to ITSB, September 4, 1991.
7. Gulden, G. K. and Reck, R. H., "Combining Quality and reengineering for Operational Superiority," Indications, September/October, 1991, Vol. 8, No. 1, pp. 1-9.
8. Hammer, M., "Reengineering Work: Don't Automated, Obliterate," Harvard Business Review, July/August 1990, pp. 104-112.
9. Hammer, M., "Why We Need Both Continuous and Discontinuous Improvement," Indications, September/October, 1991, Vol. 8, No. 1, pp. 6-7.
10. IDA, Functional Economic Analysis of DoD Functions, Users Manual, Institute of Defense Analysis, 1991.
11. Integrated Computer-Aided Manufacturing (ICAM) Architecture Part II, Vol. IV - Function Modeling Manual ($IDEF_0$)
12. Marca, D. A. and McGowan, C. L., SADT: Structured Analysis and Design Technique, New York: McGraw Hill, 1987.
13. Mayer, R. J. and Painter M. K., "The IDEF Suite of Methods for System Development & Evolution," working paper, Knowledge Based Systems Inc., 1991.
14. Robson, G. D., Continuous Process Improvement: Simplifying Work Flow Systems, New York: The Free Press, 1991.

15. Sibley, E. H., "An IDEF Family Portrait," working paper, George Mason University, 1988.

16. Sibley, E. H., "The Evolution of Approaches to Information Systems Design Methodology," in Information Systems Design Methodologies: Improving The Practice, edited by Olle, T. W., Sol, H. G., and Verrijn-Stuart, A. A., North-Holland, 1986, pp. 1- 17.

17. Yoemans, M., Functional Policies for Corporate Information Management, presentation at IDEF Users Group Symposium, October 16, 1991.

Framework for Performing Business Case Analysis (BCA) in Office of the Secretary of Defense (OSD) Staff Offices

I. Preparation

A. Obtain Upper Management Support for BCA project.
   1. Brief upper management on the BCA procedures to be used.
   2. Obtain points of contact (POCs) to support the analysis team in the following categories: Functional POCs, Information Systems (including office automation) POCs, and Organizational POCs.

B. Obtain Mission and Functions Statement of the organization(s) under study.
   1. Determine external interfaces (customers, suppliers, and competitors).
   2. Determine internal interfaces (customers, suppliers, and competitors).
   3. Determine categories and types of products and services.

C. Obtain Organization Chart of organization(s) under study.
   1. Determine manpower authorizations by position and grade/rank.
   2. Determine personnel on-board and in-hire by position and grade/rank.
   3. Determine average fully-burdened rate for each grade/rank.

D. Obtain Inventory of current and on-order office equipment and facilities.
   1. Determine authorized and on-hand equipment and facilities.
   2. Determine age and price (new) of each piece of equipment.
   3. Develop depreciated value of equipment.
   4. Develop replacement schedule for existing equipment.

E. Obtain Operating Budgets of organization(s) for previous and current year.
   1. Identify all expenditure object classes and past/present budgets.
   2. Identify other sources of organization funding or support, if any.
   3. Identify budget short-fall or surplus by object class, if applicable.

II. Abreviated BCA

A. Conduct Interviews with functional, information systems, and organizational POCs to identify functions, information systems (including office automation), and organizations, and their suspected cost drivers. For each cost driver:
   1. Ascertain time spent on cost driver per unit time (day/week/month/year).
   2. Ascertain reason for inefficiency or ineffectiveness.
   3. Ascertain suggested method for improvement.
   4. Ascertain estimated savings (time or $) per unit time.

B. Estimate Workload Factors for current year and planning years.
   1. Identify all types of work performed and measures of effectiveness.
   2. Identify sources of workload information.
   3. Identify quantity and quality of each type of work

C. Perform High-Level IDEF0 Process "As-Is" Modelling of organization.
   1. Identify macro-level inputs, outputs, controls, and mechanisms via POCs.
   2. Prepare "As-Is" A0 Context Model plus Level 1, at a minimum. (If time permits, "As-Is" model should be decomposed to a level that will identify specific business areas believed to contain potential for improvement either from process re-engineering or OA.)
   3. Validate macro-level inputs, outputs, controls, and mechanisms via POCs.

D. Map Baseline Costs to the IDA template.
   1. Management and Support Costs.
   2. Operations Costs.

   [1. Personnel and Overhead Costs.
      a. Map actual expenditures (if available) and / or current year budgets into the current FY of the IDA template.
      b. Estimate outyear costs using current office automation support and relative workload factor for remainig years in the planning horizon.
   2. OA Equipment and Facilities.
      a. Estimate outyear equipment purchases based on obsolescence schedule from Section I.C above. Consider relative workload factor for each FY.]

E. Perform High-Level IDEF0 Process "To-Be" Modelling of organization.
   1. Identify opportunities to reduce costs and improve performance based on suspected cost drivers identified above via POCs.
   2. Prepare "To-Be" A0 Context Model plus Level 1, at a minimum. ("To-Be" model should also be decomposed to the same level as the "As-Is" model. Note: If current business processes are sound and OA technology alone will provide increased productivity or cost savings, the "To-Be" model will not differ from the "As-Is" model.)
   3. Validate macro-level inputs, outputs, controls, and mechanisms via POCs.

F. Map Alternative Costs to the IDA template.
   1. Management and Support Costs.
   2. Operations Costs.

   [1. Personnel and overhead costs.
      a. Estimate impact of process re-engineering and / or OA technology on baseline costs for each FY.
      b. Consider relative workload for each FY.
      c. Map high and low estimated costs to the IDA template.
   2. OA Equipment
      a. Estimate high and low OA equipment costs supporting the alternatives for each FY. Consider relative workload factors.]

G. Perform Risk Adjusted Discounted Cash Flow (RADCF) using IDA model.

H. Determine risk acceptablity and refine estimates if necessary.

I. Perform sensitivity analysis on alternatives and rerun RADCF model.

J. Document results and brief management.

K. Iterate at next lower level of detail, if necessary.

SUPPLEMENTARY

INFORMATION

OFFICE OF THE ASSISTANT SECRETARY OF DEFENSE
6000 DEFENSE PENTAGON
WASHINGTON, D.C. 20301-6000

COMMAND, CONTROL, COMMUNICATION AND INTELLIGENCE

1994

ERRATA AD-B166 990L

151. 20/19
Targets Req'd

MEMORANDUM FOR DEFENSE TECHNICAL INFORMATION CENTER
ATTN: DATA BASE SUPPORT, WILLIAM BUSH

SUBJECT: Corporate Information Management Bibliography Limitations

We have reviewed the documents under our purview and currently limited for distribution through DTIC. We have determined that the limitations on the following documents can be changed to Catagory A, Approved for Public Release, Distribution Unlimited:

ACCESSION NUMBER:

M200225
B171442
M200146
M200149
M200146L Duplicate of item No. 3, above
M200107L
B164404
B166990L
M200140

The following documents will remain under the current Category E distribution limitations:

ACCESSION NUMBER:

B174459L
M200175L
M200176L

Connie Leonard
for Michael S. Yoemans
Director, Functional Process Improvement

cc: Frank FitzMaurice, DTIC

ERRATA - AD B166 990L